Informatique théorique III
(Automates, langages & calculabilité)

Formulaire du Cours 2004-2005
Partie 7
EPFL – I&C

Uwe Nestmann
Sébastien Briais
Daniel C. Bünzli

14 février 2005

Bibliographie

[HMU03] John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to Automata Theory, Languages and Computation*. Pearson Education International, 2003. ISBN 0321210298.

Table des matières

Chapitre 3

Langages non-contextuels

$Id: notes-7.tex,v 1.17 2004/12/01 18:03:31 sbriais Exp $

semaine 7

Voir [HMU03] §5.1–2, §5.4.

3.0.1 Notation Soit $G = (V, \Sigma, P, S)$ une grammaire.
On note parfois :

$$\alpha \to \quad \beta_1 \;\big|\; \cdots \;\big|\; \beta_n$$

pour des productions $\alpha \to \beta_1, \ldots, \alpha \to \beta_n$ dans P.

Par ailleurs on utilise les symboles $\alpha, \beta, \ldots$ pour les éléments de $(V \cup \Sigma)^*$ et $A, B, \ldots$ pour les éléments de V.

3.0.2 Lemme Soit $G = (V, \Sigma, P, S)$ une grammaire. Soit $\alpha, \alpha' \in (V \cup \Sigma)^*$.
Si $\alpha \Rightarrow_G^* \alpha'$, alors pour tout $\beta, \gamma \in (V \cup \Sigma)^*$: $\beta\alpha\gamma \Rightarrow_G^* \beta\alpha'\gamma$.

3.1 Grammaires non-contextuelles

Une grammaire non-contextuelle (c.-à-d. de type 2, voir §1.4) génère un langage dit *libre de contexte* ou *algébrique* ou *non-contextuel* (en anglais : *context-free language*).

Nous ne considérons que des grammaires $G = (V, \Sigma, P, S)$ telle que :
- si $A \in V$, alors il existe $\alpha \in (V \cup \Sigma)^*$ tel que $(A, \alpha) \in P$;
- si $(A, \alpha) \in P$, alors $\alpha \neq A$.

3.1.1 Définition (Dérivation la plus à gauche)
Une dérivation *la plus à gauche* (resp. *droite*) $\alpha_o \Rightarrow_G \ldots \Rightarrow_G \alpha_n$ est telle que pour tout $i \in [1, n]$ la dérivation directe $\alpha_{i-1} \Rightarrow_G \alpha_i$ remplace le symbole non-terminal le plus à gauche (resp. droite).

3.1.2 Lemme Soit $G = (V, \Sigma, P, S)$ une grammaire.
Si $w \in \mathrm{L}(G)$, il existe une dérivation la plus à gauche $S \Rightarrow_G^* w$.

3.2 Arbres de dérivation

3.2.1 Définition (Arbre)
On définit l'ensemble des arbres $\mathcal{T}_N$ avec nœuds N par :

$$\text{(T-FEUILLE)}\ \frac{}{(n) \in \mathcal{T}_{\{n\}}} \qquad \text{(T-NŒUD)}\ \frac{k \geq 1 \qquad t_1 \in \mathcal{T}_{N_1}\ \ldots\ t_k \in \mathcal{T}_{N_k}}{(n\ t_1 \ldots t_k) \in \mathcal{T}_{\{n\} \uplus N_1 \uplus \ldots \uplus N_k}}$$

Pour un arbre $t \in \mathcal{T}_N$, on définit :

1. la *racine* de t par :

$$\begin{aligned} \text{racine}\big((n)\big) &= n \\ \text{racine}\big((n\ t_1 \ldots t_k)\big) &= n \end{aligned}$$

2. l'ensemble des *sous-arbres* de t par :

$$\begin{aligned} \text{sous-arbres}\big((n)\big) &= \emptyset \\ \text{sous-arbres}\big((n\ t_1 \ldots t_k)\big) &= \bigcup_{i \in [1,k]} \text{sous-arbres}(t_i) \cup \{(n\ t_1 \ldots t_k)\} \end{aligned}$$

3.2.2 Définition (Arbre de dérivation) Soit $G = (V, \Sigma, P, S)$ une grammaire. Un arbre de dérivation (t, ψ) est un arbre $t \in \mathcal{T}_N$ muni d'une fonction d'étiquetage $\psi : N \to (V \cup \Sigma \cup \{\epsilon\})$ telle que pour tout $(n\ t_1 \ldots t_k) \in \text{sous-arbres}(t)$:

1. $\psi(n) \in V$.
2. Si $\psi(n) = A$ et pour tout $i \in [1, k]$ on a $\psi\big(\text{racine}(t_i)\big) = X_i$,
   alors $(A \to X_1 X_2 \cdots X_k) \in P$.
   De plus, si l'un des $X_i = \epsilon$ alors $k = 1$ (et donc $i = 1$).

3.2.3 Définition (Mot d'un arbre de dérivation) Le *mot* $\text{mot}(t, \psi)$ *généré* par un arbre de dérivation (t, ψ) est donné par :

$$\begin{aligned} \text{mot}((n), \psi) &\triangleq \psi(n) \\ \text{mot}((n\ t_1 \ldots t_k), \psi) &\triangleq \text{mot}(t_1, \psi) \cdot \ldots \cdot \text{mot}(t_k, \psi) \end{aligned}$$

Si $\alpha = \text{mot}(t, \psi)$, on dit que (t, ψ) est un *arbre de dérivation* pour α.

3.2.4 Théorème Soit $G = (V, \Sigma, P, S)$ une grammaire, $A \in V$ et $\alpha \in (V \cup \Sigma)^*$. Les deux propositions suivantes sont équivalentes :

1. Il existe une dérivation $A \Rightarrow_G^* \alpha$.
2. Il existe dans G un arbre de dérivation (t, ψ)
   avec $\text{racine}(t) = A$ et $\text{mot}(t, \psi) = \alpha$.

3.2.5 Corollaire (Arbre d'analyse) Soit $G = (V, \Sigma, P, S)$ une grammaire et $w \in \Sigma^*$. Alors $w \in \mathrm{L}(G)$ ssi il existe dans G un arbre de dérivation (t, ψ), dit *arbre d'analyse*, avec $\text{racine}(t) = S$ et $\text{mot}(t, \psi) = w$.

3.3 Ambiguïté

3.3.1 Définition Une grammaire $G = (V, \Sigma, P, S)$ est *ambiguë* s'il existe un mot $w \in \mathrm{L}(G)$ avec au moins deux arbres d'analyse distincts.

3.3.2 Théorème Il n'existe pas d'algorithme qui détermine si une grammaire est ambiguë.

3.3.3 Théorème Soit $G = (V, \Sigma, P, S)$ une grammaire et $w \in \Sigma^*$.
Les deux propositions suivantes sont équivalentes :

1. Il existe au moins deux arbres d'analyse distincts pour w.
2. Il existe au moins deux dérivations la plus à gauche distinctes pour w.

3.3.4 Définition Un langage L non-contextuel est *intrinsèquement ambigu* si toute grammaire G telle que $\mathrm{L}(G) = L$ est ambiguë.

3.3.5 Théorème Il existe des langages intrinsèquement ambigus.